Semantic-Oriented Description Techniques for Network Management Information Modeling

Deh-Min Wu

Technical University of Berlin, Institute of Open Communication Systems,
Department of Computer Science, ZAZ 520, Hardenbergplatz 2, 10623 Berlin, FR Germany,
e-mail: wu@fokus.gmd.de

Abstract

This paper proposes a semantic-oriented approach to classifying network management information so that the formal description of network states can be supported. We describe the classification concept which is slightly different from the approaches like *rule-based* or specification language *object-Z*. We apply the situation classification capability extended from those found in *KL-ONE* or *Description Logic*, in order to organize the state and event for network management in a hierarchical structure. This approach is applied in order to *formally describe information* for network management purposes.

Keyword Codes: K6.4; F4.3; H1.1

Keywords: System Management; Formal Languages; Systems and Information Theory

1. INTRODUCTION

Today the integrated network is getting larger and larger, causing the information overhead to increase and hence making common understanding more difficult than before. One example of this sort of problem is specifying the network state and events. Without any help of formal description techniques, it is difficult to understand the real state of the network, and to enable an efficient deduction of network management actions. Therefore, the first step in managing a network is to specify these monitoring data for the network manager.

Classification of information is applied to determine whether management information possesses particular characteristics so that the specification of information corresponding to these properties can be carried out. The function of ***defining an information model*** is to abstract data so that the semantics of raw data from network resources is presented to the manager. The trend for the management of future broadband networks is object-oriented. Therefore, our concept of information classification has integrated object-oriented analysis with *term classification facilities* [1], like those found in the knowledge representation language *KL-ONE* and its successor *description logics*.

D. Hogrefe et al. (eds.), *Formal Description Techniques VII*

A general question is how to model world information. Different proposals have been made in the last decade for solving this problem, e.g. Chen's *entity-relationship* model, Sowa's *conceptual graph*, and *object-oriented* concepts [6,7,8]. For distributed applications like network management we prefer the ideas set out in situation theory, which is based on *cognitive science* and *decentralized artificial intelligence* [2,3,4,5], for the following reasons:

- A situation is the partial information that is always combined with spatial, temporal data. Network events, which are recognized and reported by agents, are passed onto the network manager with their location of occurrence and sequence.
- A situation provides information about system configuration and state changes; a situation is applied as the discrete information exchanged by intelligent autonomous agents to describe the network occurrence and its environment in order to derive an action.

The most important differences between the approach as INFOMOD information exchange for open distributed processing [9] and our approach is that our analysis attempts to support complex information structures and consistent semantics. As will be discussed later, the structure of information modeled to represent a network situation is similar to the relational model. In such a model the relationships between network elements are specified by the participants associated in a relationship. Our approach differs from *Object-Z* [10] in that a situation is a description and not a schema definition. A situation may contain an individual, a variable or another situation within it. These characteristics make our proposal more flexible with complex description cases.

The application of situations to represent the network configuration is proposed and their classification is discussed. Our approach is designed for descriptive management information, where information has to be inductively defined to ensure a consistent interpretation. Such an approach is suitable for defining new specific concepts by more generic primitive concepts like those in [1]. The objectives and also the major advantages of our proposal include:

- Situation classification provides a complex structural description. We specify a network situation by a relation and objects, e.g., a relation and its participants are a data structure to be classified. Suppose a certain network configuration is represented by a concrete real situation. Then we can apply our classification algorithm to find out its situation type.
- In addition, our proposal specifies information semantically based on the embedded situational information inductively defined by more general information. Each parameter in a situation is translated to its most generic definition and these are then compared. This characteristic allows the information exchange to be achieved with a common understanding of most generic, primitive definitions, but with no obligation to the specific classes.

2. SITUATION DESCRIPTION

In the management information models all network resources are abstracted as objects and the relation between them is modeled by a relationship. We extend the understanding of object classes as concept type in order to apply the object classes and instances to represent the concepts and individuals in a situation. In description logic, concepts are assertions or descriptions about the state of the world. Individuals are particular instantiations of concepts, and roles provide a way to relate individuals. We accordingly obtain a mapping between class and concept, instance and individual, attribute and role.

Based on this management information model, we apply the situation theory of Barwise and Perry [4,5] and Devlin [3] to describe network states and events. In a situation the world is viewed as a collection of objects, properties, and relations. An ***infon*** is a unit (discrete) information item, and a ***situation*** is modeled as a set of infons f_i. An infon is defined as an associated object set, formally represented as a tuple of $f_i = <r_i, o_{i1}, o_{i2}, \ldots, o_{in}, p_i>$, where r_i is n-ary relationship, o_{ij} are objects, and p_i is the polarity of the infon. If the polarity is 1, then the infon is a positive state of affairs. The value 0 identifies it as negative. A situation is represented as a set of infons $\{f_1, f_2, \ldots, f_m\}$.

A composite situation description like

> ***Situation A:*** "A line has cable failure at 8 p.m." ⇔ *<has, line, cable failure,* l_a*, 20, 1>*

it describes a real situation happened in network at 8 p.m.. We use the letters l_a for an arbitrary location. This situational information is exchanged between agents and managers which classify this information into defined situation types, which are known or understood in a particular management domain. For instance,

> ***Situation B:*** "A connection has at least one failure at night" ⇔ *<has, connection, <atleast, 1, failures>,* l_b*, night, 1>*

is a defined situation type. We will say that situation *B* subsumes situation *A*, because a cable or a line is a physical element of a connection, and 8 p.m. is intuitively defined as time at night. In this example, the real situation *A* is categorized by a network manager with the situation class *B*.

The description logic uses concepts as the description of a data structure. This language is a set of terms built up from a set of primitive concepts and roles, using the operators *and, or, not, at-least, at-most, all* to construct more specific terms. A concept is used as an object class and a role is as a binary relationship. For instance, all terms used to represent a situation can be modeled by more generic terms in description logic in order to define their semantics.

The technique underlying it is term subsumption. This is the capability to recognize whether a term is more special than another one. This is simply the task of a network manager to classify information so that descriptive information acquires their semantics class. The situation subsumption is therefore used to classify situation descriptions and to specify them with subsumption relations.

The classification problem of a situation can be supported by the introduction of the subsumption. A is subsumed by B if the extension of A to the real world domain is contained in B. The notation $\subset$ is used for set inclusion. Mathematically, an extension function ξ is defined as

Definition 1: ξ : *description class* $\longrightarrow$ *instances*

$\xi(A) \subset \xi(B)$, *for all subordinate description class A of B*

A partial order subsumption relation $\leq$ is defined as a binary relation between situation classes A *and* B:

Definition 2: $A \leq B$, *iff* $\xi(A) \subset \xi(B)$

The problem of network management is to determine the class of a situation description, e.g. to test whether the received information can be classified by defined situation classes or not. The proposed subsumption algorithm applied to solve this problem is recursively defined to decompose a situation description. Since a situation is a set of infons, this algorithm tests all infon subsumption between two situations. In turn, the subsumption for each object/relation in infons are tested for the comparison of infons. Finally, the algorithm checks whether an inheritance specialization relation exists between two classes or not.

3. MANAGED RELATIONSHIPS AND OBJECTS

Object-oriented modeling is employed for situation description while inheriting the characteristics of the conceptual model. If objects are used to represent a concept such as a failure or a connection, we call them *conceptual managed objects* (CMO). The classification problem of these conceptual objects is similar to the term subsumption in description logic.

As defined previously, an infon consists of a relation and objects. The situation subsumption is not solved if the relation is not included into the subsumption algorithm. For instance, the relation "has" in our previous example could have the same semantics as "own" or "get". For this reason, it is necessary to define a class to group all relations with the same semantics. According to relationship classes, we can apply specialization to declare more specific relationship classes. In this manner, relation is an entity with roles referring to other objects. We call such relationships *conceptual relationship objects* (CRO).

Any CROs and CMOs used in a situation are inductively defined information to ensure a consistent interpretation. Such an approach is suitable for defining new specific concepts by more generic primitive concepts. If a relationship is added to the *Guidelines for the Definition of Managed Objects* (GDMO), a relationship in a situation is defined as an n-ary association which has to be mapped to binary relationships allowed in description logics. That is, a relationship is referred to as an entity containing referential attributes.

A relationship is specified by a relationship class, and a relationship referring to an object/ properties with roles associates with other objects/properties. An attribute in a CMO definition is replaced by a role in that of a relationship. Applying the relationship as a special type of MO, a situation can be defined and classified for exchanging in a distributed environment. As is at present conceived by a joint CCITT-ISO group [7], a template for the structure of a relationship class may be as follows:

```
<relationship-class-label>    RELATIONSHIP CLASS
          [DERIVED FROM <relationship-class-label>
                [,<relationship-class-label>]*; ]
          [BEHAVIOUR <behaviour-definition-label>
                [,<behaviour-definition-label>]*; ]
          [ROLE <role-label> role-properties
                [REGISTERED AS object-identifier];] *
REGISTERED AS object-identifier;
```

To identify CMO and CRO in different open management systems with common understanding of their values, their classes are specified and managed in inheritance, containment and registration trees. Inheritance helps us derive new object classes from an existing one, and a containment tree declares object dependency/containing relations. A registration tree provides a global unique object identifier for naming an object in open systems. We identify a CMO/CRO by $i::C$, where C is the class of an object i. We apply $*::C$ (or simply C) to denote an arbitrary element in class C.

4. Subsumption Algorithm

We apply set theory for the extension function in the description logic to the definition of the subsumption rules. We use the letters C to denote a CMO/CRO class (a set), f an infon (information unit) and i an object instance (a set element). Before we introduce subsumption, we define semantics equivalence for class/instance of conceptual objects:

Definition 3: Two classes C, C' are equivalent ($C \cong C'$) iff $\xi(C) = \xi(C')$.

Definition 4: If two instances $i::C$ and $i'::C'$ have the same instance values, and $\xi(C) \subset \xi(C')$ *or* $\xi(C') \subset \xi(C)$, then i and i' are equivalent ($i \cong i'$).

To check if two object instances are semantically equivalent, we only need to compare the values involved in these instances. This can be demonstrated as the case of having information sent twice to the manager. How should we define a case where two situation descriptions are semantically equivalent for network managers so that duplicated events for one state change can be ignored. From these two definitions we have:

Proposition 1: CMO/CRO $O = *::C$ and $O' = *::C'$, O and O' are equivalent iff $C \cong C'$. * is a wild-card.

Proposition 2: CMO/CRO $O = i::C$ and $O' = i'::C'$, O and O' are equivalent iff $i \cong i'$.

The subsumption of infons will be carried out by the following steps. Two relation notations are used:

- relation $X :< Y$, where X and Y are classes, is true when class X is a direct subordinate class of Y in an inheritance hierarchy.
- relation $i := Y$ is true when i is an instance of class Y.

We apply the following propositions to subsume conceptual objects C, C' and C''. The intersection operation is denoted as $\cap$ and implication as $\Rightarrow$.

Proposition 3: $C :< C' \Rightarrow C \leq C'$

Proof: Because C is derived from the class C', the extension of C in the real world is also an extension of C', i.e. $\xi(C) \subset \xi(C')$.

Proposition 4: $C :< C'$ *and* $C' \leq C'' \Rightarrow C \leq C''$

Proof: Applying Proposition *3*, $C :< C' \Rightarrow C \leq C'$. When we apply the transitive property of the relation $\leq$, then $C \leq C'$ and $C' \leq C''$ implies $C \leq C''$.

Proposition 5: $i := C$ *and* $C \leq C' \Rightarrow i := C'$

Proof: $i := C \Rightarrow i \in \xi(C)$,

$C \leq C' \Rightarrow \xi(C) \subset \xi(C')$, i.e., $i \in \xi(C')$

Proposition 6: $C \leq C'$ *or* $C \cong C' \Rightarrow i :: C \subset * :: C'$, where * denotes a wild-card variable.

Proof: Suppose $\exists\, i \in \xi(C)$, we know that $\xi(C) \subset \xi(C')$

that implies $i \in \xi(C')$

Proposition 7: $C \leq C' \Rightarrow * :: C \subset * :: C'$

Proof: Applying proposition *5*, $\forall\, i \in \xi(C)$ and $\xi(C) \subset \xi(C') \Rightarrow i \in \xi(C')$

In our proposed subsumption algorithm, the set operations intersection and union are applied to support the description operators such as *and, or, atleast,* etc. Its theoretical background is outside the scope of this paper and can be referenced in object-centered modeling such as [1]. Our proposal applies the subsumption concept to the situation classification. Notice that the classification of a situation is based on the comparison of set inclusion between the extension of two descriptions.

5. SITUATION SUBSUMPTION

We apply the propositions described in the previous section for the subsumption propositions for CMO/CRO. Suppose given CRO a_1, a_1' and CMO a_i, a_i' associated in infons *f* and *f'*, we know that there is an equivalence or a subsumption relation between *f* and *f'* if one of the following conditions is satisfied:

Proposition 8: For infons, $f = <a_1, a_2, ..., a_n>$ and $f' = <a_1', a_2', ..., a_n'>$, $f \cong f'$ iff $a_i \cong a_i'$, where $i = 1, ..., n$.

Proof: Two infons are equivalent when all their contained CMOs/CROs are equivalent.

For instance, a controller C_i reports that the observation *situation A* is detected, which is given in section 2. Then the composite situation *C*

Situation C:: *"Controller C_i observes situation A at 8 p.m."* $\Leftrightarrow$ *<observes, C_i::Controller, A::Situation, l_c, 20, 1)*

is equivalent to the situation *D*, iff the situation *A* is replaced by its definition.

Situation D: *"Controller C_i observes "A line has cable failure at 8 p.m." "* $\Leftrightarrow$ *<observes, C_i::Controller, <has, line, cable failure, l_a, 20, 1>, l_c, 20, 1)*

Proposition 9: For infons, $f = <a_1, a_2, ..., a_n>$ and $f' = <a_1', a_2', ..., a_n'>$, $f \leq f'$ iff there exists at least one i with $a_i \leq a_i'$ and for all others $a_j \cong a_j'$, where $j = 1, ..., n, j \neq i$.

Proof: Infon *f* is more special than infon *f'* if the relation and all object participants in *f* are either (at least once) more special than or equivalent to them in *f'*.

Proposition 10: For infons, $f = <a_1, ..., a_i, b_1, b_m, a_{i+1}, ..., a_n>$ *and* $f' = <a_1', ..., a_i', a_{i+1}' ..., a_n'>$, $f \leq f'$ iff either $a_i \leq a_i'$ *or* $a_i \cong a_i'$, where $1 \leq i \leq n$, and b_j arbitrary objects, $1 \leq j \leq m$.

Proof: Infon *f* is more special than infon *f'* if the *n* objects $a_1, a_2, ..., a_n$ in infon *f* are more special than or equivalent to that part $a_1', a_2', ..., a_n'$ of *f'*, and the roles participants in *f*, i.e. $b_1, ..., b_m$, are additional restrictions to *f*.

Now we have to make some propositions normally allowed in situation theory, but conventionally not permitted in the *object-oriented model* and *object-Z*. Infons may involve infons as their parameters, e.g. a conceptual object may occur buried in one or more levels inside an infon.

Proposition 11: For infons f and f', $f \le f'$ iff $f = <a_1, a_2, ..., f', ..., a_n>$

Proof: An infon f' is more general than an infon f when f' is contained in f.

In *Proposition 11*, f' is an embedded situation which is specified by an indirect description, as the example in C that $C \le A$ satisfies.

Proposition 12: For infons $f=<a_1,a_2,..., a_n, a_{n+1}, ..., a_m>$ *and* $f' = <a_1',a_2',...,a_n',s>$, and a situation $s = \{f_1, ..., f_k\}$, $f' \le f$ iff both of the following are satisfied:

- $\forall\ a_i, 1 \le i \le n$, either $a_i' \le a_i$ *or* $a_i \cong a_i'$
- $\forall a_i, n+1 \le i \le m, \exists f_j = <...,b_l, ...>$ that $b_l \cong a_i$ or $b_l \le a_i$, where $1 \le j \le k$

Proof: The first condition checks the subsumption relation of their n objects $a_1,a_2,..., a_n$ according to Proposition *10*. The second condition ensures that all the remaining objects in f are contained in any infons of s, i.e. for all CMOs in f, we always find a more specific CMO in s corresponding to it.

To show an application to the *proposition 12* a situation E is given,

Situation E: *"A controller observes a failure at 8 p.m."* ⇔ *<observes, *::Controller, *::failure,* l_c*, 20, 1>*

The situation E subsumes the situation C as situation C specifies a more complete description than that in E. The situation E only states that a failure is observed but the situation C specifies exactly where this failure is manifested. We further give the following propositions for situation subsumption, however, they are only used to ensure that all situation infons are subsumed by infons in another situation:

Proposition 13: $S = \{f_1,..., f_n\}$ and $S' = \{f_1',..., f_n'\}$, $S \cong S'$ iff $\forall f_i$ in S, $\exists f_j'$ in S' that satisfies $f_i \cong f_j'$, and vice versa.

Proposition 14: $S = \{f_1,..., f_n\}$ and $S' = \{f_1', ..., f_m'\}$, $S \le S'$ iff $\forall f_j'$ in S', $\exists f_i$ in S that satisfies either $f_i \cong f_j'$ or $f_i \le f_j'$

Proof: We apply the inductive approach to prove this proposition. In the case of m=1: Suppose $S' = \{f_1'\}$ and $\exists f_l$ in S that satisfies either $f_l \cong f_1'$ or $f_l \le f_1'$, then $S=\{..., f_l, ...\} \le \{..., f_1,...\} \le \{f_1\} = S'$ is satisfied.

In the case of m=k-1, *Proposition 14* satisfies: $S \le \{f_1', ..., f_{k-1}'\} = S'$. We try to prove whether m=k is also satisfied. That is to prove $S \le S'' = \{f_1', ..., f_{k-1}', f_k'\}$. As the pre-condition of *Proposition 14,* for f_k' in S'' we find a f_l in S that $f_l \le f_k'$, and $S=\{f_1,...,f_l,...f_n\}\le\{f_1', ..., f_{k-1}', f_l\}\le\{f_1', ..., f_{k-1}', f_k'\}= S''$.

By induction, for all $k=1,...,m$ that *Proposition 14* satisfies.

Notice that the following case is in general not decidable: Suppose there is one infon f_k' in S', and we cannot find any f_l in S that $f_l \le f_k'$ *or* $f_l \cong f_k'$, then $S \le S'$ cannot be determined automatically in our study. As an example of the above propositions, we give two situations $F = \{A, C\}$, $G = \{B, E\}$, i.e.

Situation F: *A=<has, line, cable failure,* l_a*, 20, 1> and C=<observes,* C_i*::Controller, A::Situation,* l_c*, 20, 1>*

Situation G: B=*<has, connection, <atleast, 1, failures>,* l_b*, night, 1> and E=<observes, *::Controller, *::Failure,* l_c*, 20, 1>*

To recognize if the situation G has a more general class than that of situation F, we check whether G subsumes F. This means proving the fact that both infons B and E in G subsume one infon in F. Since $A \le B$ *and* $C \le E$, hence the situation G subsumes the situation F is obtained.

Suppose the domain of all situations in the real world is the set SIT, then the situations S and S' are elements from this domain. As defined in the previous section, situation S is subsumed by S' iff $S \le S'$. We define a function SSUB for situation subsumption:

Definition 5: SSUB: SIT $\times$ SIT $\rightarrow$ { *true, false* }

$$\mathrm{SSUB}(S, S') = \begin{cases} \text{true, iff } S \le S', \\ \text{false, } S \not\le S' \end{cases}$$

If a real situation S is subsumed by a situation class S', then S' contains more generic information than S or S is more specific. We can specify all descriptions for network states and events using situation classes. In our previous example, a real situation C is subsumed by a more general situation E. The type of a real situation description is then classified, e.g. a real situation C has the situation description class E.

Suppose a real situation S is classified by classes in a situation hierarchy, then more general situation classes, say σ, are found as classes of this real situation description S. The most specific one in σ is defined as the (most specific) situation class of S.

In other words, all descriptions subsumed by a situation description class are defined as real situations for that class. In this way, the network information is organized by the situation subsumption function SSUB, i.e. the proposed algorithm provided by the SSUB function compares the extensions of two given situations. All description classes are structured into a hierarchy according to their subsumption relations.

The ISO/OSI structure of management information employs an object-oriented concept to model managed information. According to this object-oriented concept, classification of information is the pre-condition to define a model for information exchange in open systems. If the network situation descriptions have to be included into CMIP (*Common Management Information Protocol* [12]), we have to provide a classification method for organizing them in an object-oriented approach. In this case, the proposed classification of situation descriptions using the subsumption function SSUB proposed in this paper can be applied to structure description information for CMIP.

6. EVALUATION OF APPLICATIONS

This proposal begins with the conventional approach GDMO used by CMIP and tries to integrate description facilities into it. Situation theory and description logics which are mathematically well-founded are applied to support the proposal. The situation theory abstracts the state of affairs with mathematical formula. We apply the description logics to describe a situation so that situation theory can be represented in CMOs and CROs. That is to say, the realization of situation theory is solved by description logic.

The propositions provide the subsumption algorithms for the situation classification. Because a situation is described by recursive composition of CMOs and CROs, it makes the classification more difficult than that of CMOs alone. As described, a relationship is formally specified with a CMO-like structure and the classification of CROs is also defined. Therefore, the CRO can be integrated into GDMO and the similarity of their structure simplifies the classification of situations.

For open information modeling the classification of information is the most basic pre-condition. The situation theory has still no open information model for its application in network management. This proposal supports us to construct one for situation theory. Because CRO may be considered as an extension to GDMO, and the classification of composition of CMOs and CROs can be clarified by the subsumption algorithms, hence the classification of information in situation description is achieved. The model of situational information is therefore constructed for the pre-condition of information exchange in open environment.

For problem solving using current GDMO, we may solve a problem in the framework of an object-oriented method. For instance, suppose an object is defective, the handling of this fault has to be stored in the class information of this object. A current GDMO solution is to specify the behavior and actions in text for the handling of managed objects. No formal description techniques are applied for the specification of such information. The major benefits of such situational modelling for a managing application is to support the definition of the behavior and actions. One possible approach is to specify the behavior by a rule.

As an example for a behavior rule we may specify it formally as:

Situation H: *"if the situation H and situation A are true, then apply action T"* ⟺ *<then, <and, H::Situation, A::Situation,* l_h*,* t_h*::Time, 1>, <apply, managing system, T::Action,* l_h*,* t_h*::Time, 1>,* l_h*, *::Time, 1>*

Action T: *"go step* S_1 *and* S_2*, if successful, then go step* S_5*, else go step* S_9*"* ⟺*<go, <then, <are, <go,* S_1*::Step,* S_2*::Step,* l_t*,* t_t*, 1>, successful,* l_t*,* t_t*, 1>,* S_5*::Step,* S_9*::Step,* l_t*,* t_t*, 1>,* l_t*,* t_t*, 1>*

7. SCENARIO EXAMPLES AND IMPLEMENTATION

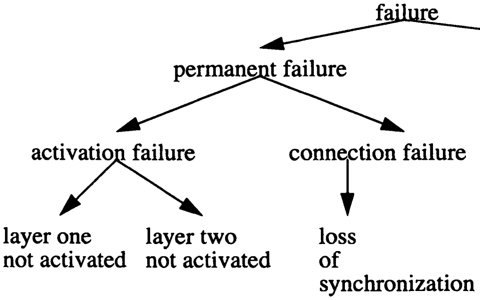

Figure 1 : CMO class inheritance hierarchy for failures

In this section we depict a part of our management information model as an example to illustrate how this proposal is applied in practice to specify monitoring information for network management. As described, we model the network elements as CMOs. CMO classes specified for network failures are illustrated in Figure 1, and for network elements and their configurations are depicted in Figure 2. In these two figures, a subordinate node is a more specific CMO class than its superior.

The CROs are also modeled like CMOs. The relations representing a situation are structured in an inheritance hierarchy. For instance, the relation class *has-parameter* is more specific than *has*. After defining these conceptual objects, we derive a situation hierarchy according to the subsumption function as depicted in Figure 4. For better understanding, we describe the situations in Figure 4 in the text. Each subordinate situation is subsumed by its superior. For example, the situation descriptions $S_{1,1,1}$ and $S_{1,1,2}$ are represented formally as

$S_{1,1,1}$ = *<has,*
networkElement,
failure>

$S_{1,1,2}$ = <*has*,
<*hasParameter, subscriberLine*,
<*is, lineOperationalState, X*> *and*
<*is, lineAdministrativeState, Y*>>,
activationFailure>

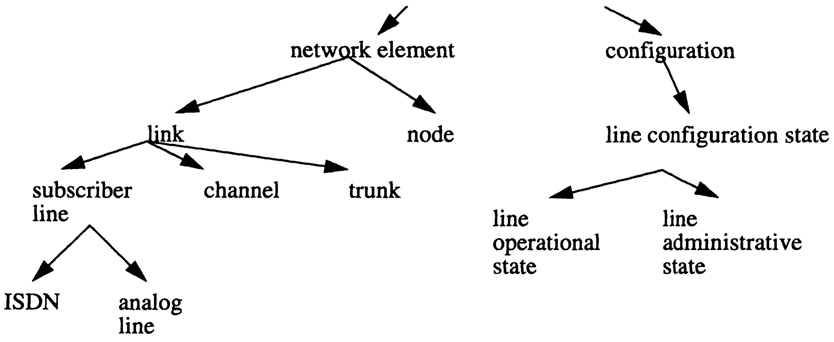

Figure 2 : CMO class inheritance hierarchy for network physical elements and logical parameters

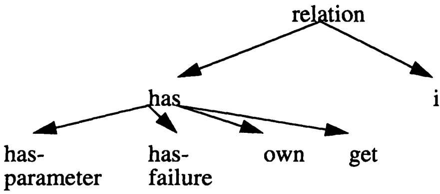

Figure 3 : CRO inheritance hierarchy

In addition, we apply SSUB to classify a network real situation. For example in Figure 4 a real situation $S_{1,1,2}$ is subsumed by situation $S_{1,1,1}$, i.e. $S_{1,1,2} \leq S_{1,1,1}$. The SSUB tests if the *network element* class and *failure* class in $S_{1,1,2}$ are more specific than those in $S_{1,1,1}$. The following subsumption relationships are proved in order to reason the validity of $S_{1,1,2} \leq S_{1,1,1}$.

subscriberLine $\leq$*networkElement.*

<hasParameter, subscriberLine, lineConfigurationState>
$\leq$ *subscriberLine.*
<is, lineOperationalState, X> and <is, lineAdministrativeState, Y>
$\leq$ *lineConfigurationState.*
activationFailure $\leq$ *failure.*

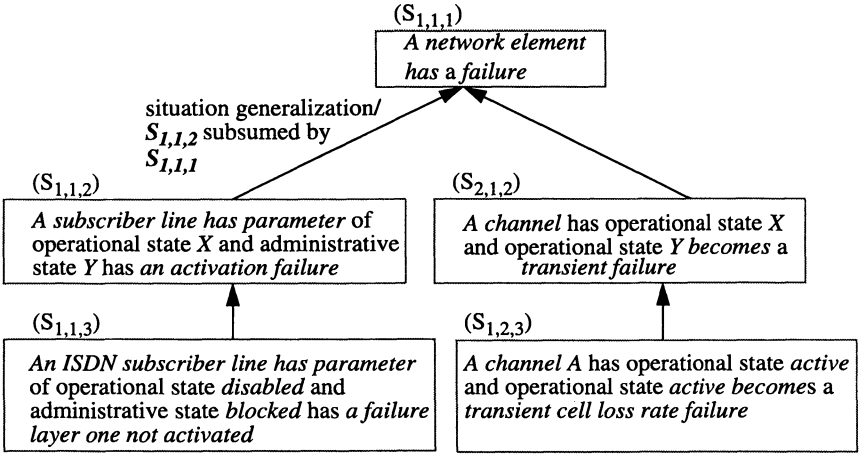

Figure 4 : An example in a digital switching system for situation hierarchy organized with the help of the subsumption function

We have used the tool *DAMOCLES* [13], which is being developed by GMD Fokus within the *Berkom* project, to maintain all conceptual object classes. *DAMOCLES* enables us to define and to administer managed object classes interactively. It can be used to check the syntactical correctness and the consistency of an information model according to the *GDMO* [11]. For the subsumption of conceptual objects, all managed objects defined in *DAMOCLES* are translated into concept definitions in the *BACK* system [14], which is developed at the TU Berlin within the framework of an *ESPRIT* project for concept management. *BACK* is based on *description logic*, which is realized in Prolog and supports the concept subsumption facilities.

8. CONCLUSION

This paper describes an extended definition of the term subsumption capability in order to describe network situations formally. This proposal puts forward the situational modeling realized by [15] so that these subsumption propositions are suitable for situational description in an open environment. The most important characteristic of our proposal is: Situation classification supports structuring of situation hierarchy. In addition, our proposal specifies information semantically, based on the embedded situational information recursively defined by a more general description.

REFERENCES

1. Bernhard Nebel, “Reasoning and Revision in Hybrid Representation Systems”, Lecture Notes in Artificial Intelligence 422, Edited by J. Siekmann, Springer-Verlag, 1990
2. Eric Werner, "Distributed Cooperation Algorithms", Decentralized A.I. Editors Y. Demazeau J.-P. Müller, North-Holland, 1990
3. Devlin, Keith J., “Logic and Information”, Cambridge University Press, 1991
4. Jon Barwise, John Perry, “Situationen und Einstellungen: Grundlagen d. Situationssemantik”, de Gruyter, 1987
5. Jon Barwise, “The Situation in Logic”, CSLI, Lecture Notes 17, 1989
6. Peter Pin-Shan Chen, “The Entity-Relationship Model - Toward a Unified View of Data”, ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976, pp. 9-36
7. ISO/IEC N7126, “General Relationship Model - Third Working Draft”, ISO/IEC JTC 1/SC 21/WG 4 Meeting, Ottawa, May 1992
8. J. F. Sowa, “Conceptual Structures - Information Processing in Mind and Machine” Reading/MA: Addison Wesley, 1984
9. J.J. van Griethuysen, “Enterprise modelling, a necessary basis for modern information systems”, MMJ Information Engineering, Casteren - The Netherlands, Convener ISO/IEC JTC1/SC21/WG7 - ODP
10. David Carrington, David Duke, Roger Duke, Paul King, Gordon Rose, and Graeme Smith, “Object-Z: An Object-Oriented Extention to Z”, Formal Description Techniques, II, Ed. S.T. Vuong, Elsevier, IFIP, 1990
11. ISO/IS 10165-4, “Information processing systems - Open systems interconnection (OSI) - Structure of Management Information”, Part 4: Guidelines for the Definition of Managed Objects, May 1992
12. Information Processing Systems - Open Systems Interconnection - Common Management Information Protocol - Part 1’, 1990
13. Marcus Wittig, Martin Pfeiler, "A tool supporting the Management Information Modeling process", IFIP Transactions Integrated Network Management III, Editor H.-G. Hegering and Y. Yemini, pp739- 750, 1993
14. Thomas Hoppe, Carsten Kindermann, J. Joachim Quantz, Albrecht Schmiedel and Martin Fischer, “BACK V5: Tutorial & Manual”, Technische Universität Berlin, Projekt KIT-BACK, March 1993
15. Deh-Min Wu, “TMN Performance Assurance using Situation Modelling on ATM-Networks”, Technical Report 1993/19 of the Department of Computer Science, Technical University of Berlin, July 1993